**SQL vs NoSQL**

**What is SQL?**

SQL, which stands for Structured Query Language, is a domain-specific programming language (e.g., a language targeted to a specific task or problem) that is commonly used for tasks such as inserting, updating, querying, and deleting data within a database. SQL is also used to create and modify database schemas (e.g., data formatting rules, table/index structure ) as well as define database access and administration parameters.

**What is NoSQL?**

NoSQL, which stands for Not only SQL, is a database management system approach used to ingest, store, and retrieve unstructured data and semi-structured data within a database. This means that data that cannot be analyzed or counted through traditional relational databases (e.g., SQL) can remain in its native format and be ingested into a NoSQL database. The reason it is called NoSQL is to emphasize that these databases can handle non-tabular, non-relational data models as well as support SQL-like query languages.

**What is structured data?**

Structured data is data that is organized in a consistent, predefined format and often consists of alphanumeric characters. Examples include financial transactions, inventory records, or customer lists which are often stored in SQL databases (e.g., relational databases).

**What is unstructured data?**

Unstructured data is data that doesn't have a predefined data model or consistent organization. In addition, unstructured data, such as social media posts, can update and change rapidly while structured data, such as bank transactions, have a much lower rate of change. Examples of unstructured data include pictures, audio files, videos, and maps.

**What is SQL database?**

When the term "SQL database" is used, it refers to a type of database where SQL is the primary programming language used to create and manage that database. SQL application programming interfaces (APIs) contain groups of functions that enable developers to execute and manage database operations without having to create individual SQL commands over and over.

Regardless of whether a SQL database is used to store transactions for a retailer or financial information for a corporation, SQL databases fall under a type of database referred to as relational databases.

**What is NoSQL database?**

NoSQL databases are databases that utilize a flexible schema that accommodates unstructured data and semi-structured data while also utilizing a non-tabular data storage method.

The use of a flexible schema enables NoSQL databases to ingest unstructured data in its native format (e.g., .txt, .JPG, MP3), which is not possible with SQL databases due to the requirement that all data align to a predefined format. Further, when NoSQL databases store data, flexible data models are employed so that unstructured data files can have different data structures and still be stored within the same collection.

**SQL database use cases**

**Regulatory compliance**

* Because SQL database structure lends itself to ACID compliance, they are often used for the purposes of storing data that must meet certain governmental or industry standards.
* The use cases below all involve some level of ACID compliance.

**Transactional databases**

* Transactional databases store data that results from an interaction between two or more parties.

Examples:

* Retailer point-of-sale (PoS) databases
* Healthcare prescription and order databases
* Commercial banking
* Accounting and financial recordkeeping databases

**Enterprise resource planning (ERP) systems**

* ERP systems are used to help businesses manage processes that are key to operations, employee management, production, and more.

Examples:

* Human resources databases
* Supply chain management systems
* Risk management systems

**NoSQL databases use cases**

**Transactional databases**

* Transactional databases can also be supported by NoSQL databases in that they are used to store unstructured data that results from an interaction between two or more parties.

Examples:

* Healthcare patient files requiring non-relational database capabilities (e.g., patient records, x-ray/scan photos and videos)
* Insurance case files (e.g., auto accident photos, injury documentation)
* Legal document databases (e.g., depositions, pleadings, case files)

**Document databases and digital asset management (DAM)**

* Document databases and digital asset management store and manage documents, images, multimedia content, videos, and more.

Examples:

* Online libraries (e.g., legal libraries, online Library of Congress)
* Digital publishing platforms such as Kindle or Nook
* Media streaming services such as Netflix and Hulu
* Online photo-sharing platforms such as Instagram or Meta

**Graph and network analysis**

* Graph and network databases are excellent at managing such data structures as recommendation engines, social networks, and associated network analysis given their ability to identify and analyze nonintuitive relationships within interconnected data elements.

Examples:

* Social network analysis (e.g., post and user metrics)
* Fraud detection which isolates unusual transactions or other anomalies
* Knowledge graphs (e.g., Gartner product/service quadrants)

**Features of MongoDB**

MongoDB is a scalable, flexible NoSQL document database platform designed to overcome the relational databases approach and the limitations of other NoSQL solutions. MongoDB is well known for its horizontal scaling and load balancing capabilities, which has given application developers an unprecedented level of flexibility and scalability.

**Key features**

1. **Document Model** – data stored in documents, group of documents is collections.
2. **Sharding** – splitting large data across multiple instances to support performance.
3. **Replication** – duplicating data by deploying multiple servers like horizontal scaling.
4. **Authentication** – to ensure only authorized users can access database.
5. **Database Triggers** – act based on specific events like execute some condition when new data is inserted.
6. **Time Series Data** – stores time series data in collection of documents with timestamp and value.
7. **Ad-Hoc Queries** – is short lived command which depend on a variable.
8. **Indexing** – improve search speed and performance by storing a specific field.
9. **Load Balancing** – to handle multiple read and write requests for the same data efficiently.